believed to exist in humans in the form of viscoelastic properties of muscles and spinal reflex loop. The concept of an inverse model is also attractive for analysis problems of biomechanical quantities, whereby internal loads are estimated from kinesiological measurements. The approach adopted here in developing a computational model of human sensory motor control is based on the concept of an inverse model coupled with nonlinear feedback (**FIG. 3**). This mechanism is compelling from the standpoint of biomechanical analysis of human motion as well as the synthesis of artificial control. Let q_d represent the desired kinematics, obtained from motion capture data. The following control law ($D\tau'$), when applied to the system equations, will result in a simulated response that will track and reproduce the desired kinematics data,

$$D\tau' = J(q)\ddot{q}^* + B(q, \dot{q})\dot{q} + G(q) + T_{ad} - \frac{\partial C^T}{\partial q}\Gamma' \tag{5}$$

[**0047**] where,

$$\ddot{q}^* = a\ddot{q}_d + K_p(q_d - q) + K_v(\dot{q}_d - \dot{q}) \tag{6}$$

[**0048**] The diagonal matrices K_p and K_v represent the position and velocity feedback gains, respectively. The eigenvalues of the closed loop system are related to the feedback gains by the following,

$$K_p = -(\lambda_1 + \lambda_2) \tag{7}$$

$$K_v \lambda_1 \lambda_2 \tag{8}$$

[**0049**] A critically damped response (fastest possible non-oscillatory response) to the tracking error can be achieved by specifying the eigenvalues to be equal, real, and negative. The parameter a is constant and set to 0 or 1, depending on the severity of noise in the measurements. If the desired trajectories are obtained from noisy motion capture measurements, it may be appropriate to set a=0 and to specify the eigenvalues to be large and negative. This way, tracking is achieved without the need to compute unreliable accelerations from noisy kinematics data.

[**0050**] Muscle Force and Muscle Capacity

[**0051**] The muscle force and muscle capacity module should ideally be implemented in the forward path of the closed loop system (as shown in **FIG. 5**). However, it may also be implemented as a separate module whose output is used for analysis purpose only. In the latter case, the module's inputs would tap into the required variables of the closed loop system, but the module would not alter the closed loop dynamics.

[**0052**] In either case, a number of different muscle force distribution algorithms may be implemented. The underlying concepts of our choice of muscle force distribution algorithm is presented below.

[**0053**] The relationship between the net muscular moment τ_m and the muscle forces F_m is given by,

$$D\tau_m = -\frac{\partial L^T}{\partial q} F_m \tag{9}$$

[**0054**] where L is the overall length of the muscle actuator, and $\partial L^T/\partial q$ is an (n×m) muscle moment arm matrix. Since the number of muscles (m) exceeds the degrees of freedom (n), the computation of the muscle actuator's excitation inputs (and the resulting forces) from an inverse dynamics computation amounts to solving a problem that is inherently ill-posed. Static, nonlinear optimization has been used extensively to predict the individual muscle forces to produce the required torque. There are several compelling reasons for using static optimization to predict the individual muscle forces: first, static, non-linear optimization techniques have well developed theoretical foundations. With the advance of commercial software for solving general, constrained, multi-variable non-linear optimization problems, it is now possible to solve sophisticated problems numerically in relatively short time. Second, the notion that muscle forces are controlled in some way to optimize physiological criteria has great intuitive appeal. It has been shown that for motions like walking, static optimization yields very similar results to dynamic optimization (Anderson, F C and Pandy, M G., Static and Dynamic Optimization Solutions for Gait are practically equivalent, Journal of Biomechanics 34, 2001, 153-161, 2001).

[**0055**] A muscle force and muscle capacity module takes the computed torques from the inverse model (denoted by $D\tau'$) as inputs and calculates the muscle forces based on a static optimization criterion (module **410** in **FIG. 4**). While any cost function can be defined in solving the optimization problem, the one used here minimizes the sum of muscle activations squared

$$J = \sum_{i=1}^{m} a_i^2 \tag{10}$$

[**0056**] where m is the number of muscles crossing the joint, a_i is the activation level for muscle i and is constrained to be between 0.01 and 1.0. A muscle force F_i for muscle i can be represented as below;

$$F_i = a_i F_i^0 \tag{11}$$

[**0057**] where

$$F_i^0$$

[**0058**] represents a maximum force limit for muscle i. A gradient based technique can be used to numerically solve for the muscle activations that minimize the cost function J while satisfying the joint moment equilibrium for all degrees of freedom of interest. The optimization problem can be solved using constrained nonlinear optimization (Sequential Quadratic Programming; AEM Design). Once the muscle activations are obtained, the muscle force can then be determined using the force-length-velocity-activation relationship of muscle (Zajac, F. E. Muscle and tendon: Properties, models, scaling, and application to biomechanics and